



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71220895
Nama Lengkap	Drestanta Dipta Jalu Prakasya
Minggu ke / Materi	13 / Fungsi Rekursif

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Fungsi ini berulang dengan pola terstruktur dan memiliki dua bagian penting: Base Case, yang menentukan kapan fungsi berhenti, dan Rekursif Case, yang berisi pernyataan yang terus diulang hingga mencapai Base Case. Fungsi rekursif harus digunakan dengan hati-hati agar tidak menyebabkan unlimited loop dan menghanguskan program.

MATERI 2

Fungsi rekursif memiliki kelebihan yaitu kode program yang lebih singkat dan elegan, serta kemampuan untuk memecahkan masalah kompleks dengan membaginya menjadi submasalah yang lebih kecil. Namun, kelemahan fungsi rekursif adalah penggunaan memori yang lebih besar karena setiap pemanggilan membutuhkan ruang memori tambahan, pengorbanan efisiensi dan kecepatan, serta kesulitan dalam debugging dan pemahaman.

MATERI 3

```
1     def function_name(parameter_list):  
2         ...  
3         function_name(...)  
4         ...
```

Fungsi rekursif dapat diubah menjadi solusi iteratif. Solusi iteratif untuk menghitung faktorial adalah dengan menggunakan loop untuk mengalikan angka dari 1 hingga n dan menyimpan hasilnya dalam variabel total. Solusi iteratif umumnya lebih efisien dan memiliki performa yang lebih baik daripada solusi rekursif.

$$\text{fact}(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot \text{fact}(n - 1) & \text{if } n > 0 \end{cases}$$

Pseudocode (recursive):

```
function factorial is:
input: integer  $n$  such that  $n \geq 0$ 
output:  $[n \times (n-1) \times (n-2) \times \dots \times 1]$ 

    1. if  $n$  is 0, return 1
    2. otherwise, return  $[n \times \text{factorial}(n-1)]$ 

end factorial
```

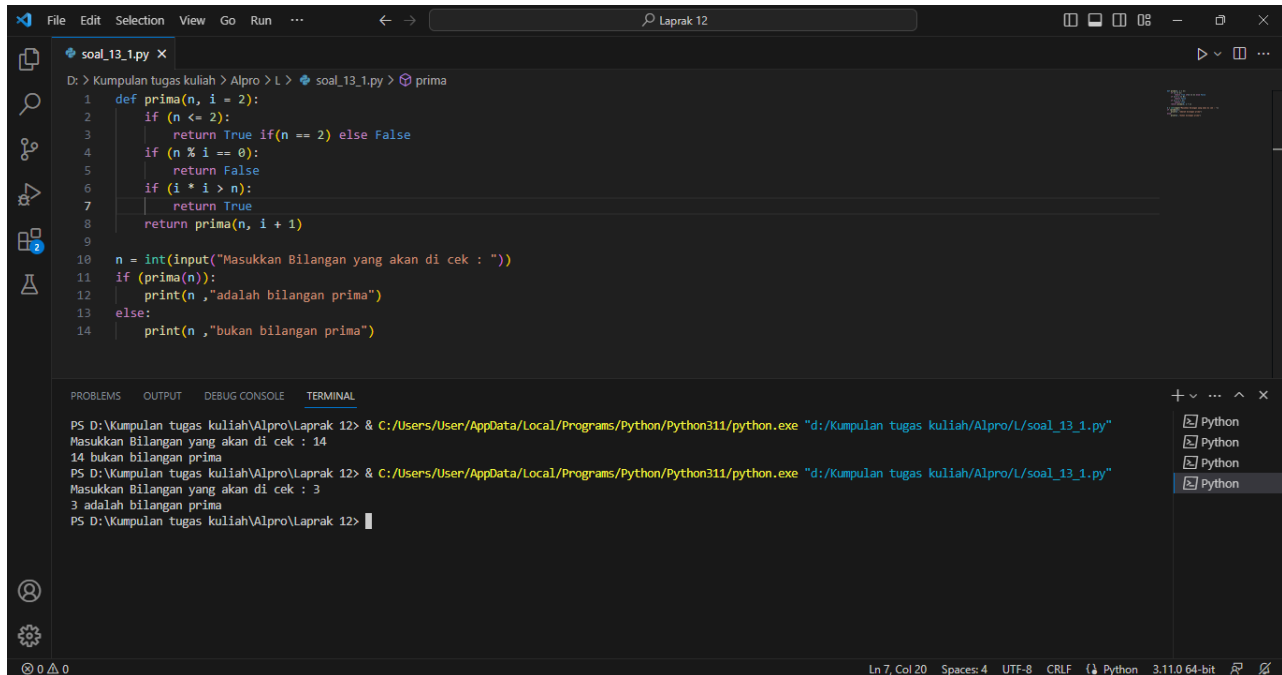
```
1  def faktorial(n):
2      if n==0 or n==1:
3          return 1
4      else:
5          return faktorial(n-1) * n
6
7  print(faktorial(4))
```

```
1.
2. calc_factorial(4)           # 1st call with 4
3. 4 * calc_factorial(3)      # 2nd call with 3
4. 4 * 3 * calc_factorial(2)  # 3rd call with 2
5. 4 * 3 * 2 * calc_factorial(1) # 4th call with 1
6. 4 * 3 * 2 * 1              # return from 4th call as number=1
7. 4 * 3 * 2                  # return from 3rd call
8. 4 * 6                      # return from 2nd call
9. 24                        # return from 1st call
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal -soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1



```
File Edit Selection View Go Run ... Laprak 12
soal_13_1.py x
D: > Kumpulan tugas kuliah > Alpro > L > soal_13_1.py > prima
1 def prima(n, i = 2):
2     if (n <= 2):
3         return True if (n == 2) else False
4     if (n % i == 0):
5         return False
6     if (i * i > n):
7         return True
8     return prima(n, i + 1)
9
10 n = int(input("Masukkan Bilangan yang akan di cek : "))
11 if (prima(n)):
12     print(n, "adalah bilangan prima")
13 else:
14     print(n, "bukan bilangan prima")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Kumpulan tugas kuliah\Alpro\Laparak 12> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "d:/Kumpulan tugas kuliah/Alpro/L/soal_13_1.py"
Masukkan Bilangan yang akan di cek : 14
14 bukan bilangan prima
PS D:\Kumpulan tugas kuliah\Alpro\Laparak 12> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "d:/Kumpulan tugas kuliah/Alpro/L/soal_13_1.py"
Masukkan Bilangan yang akan di cek : 3
3 adalah bilangan prima
PS D:\Kumpulan tugas kuliah\Alpro\Laparak 12> |
```

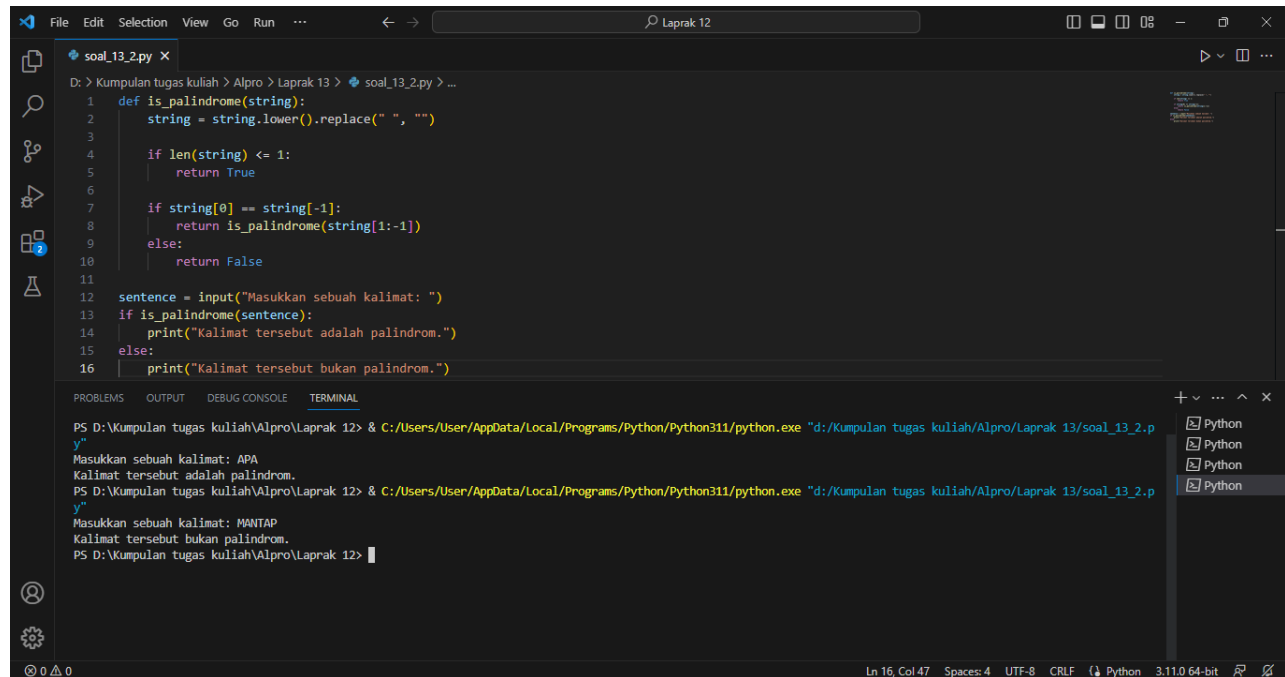
Dalam program di atas, kita menggunakan fungsi rekursif “*prima*” untuk menentukan apakah suatu bilangan *n* adalah bilangan prima atau bukan. Fungsi tersebut memiliki dua argumen: *n* adalah bilangan yang akan dicek, dan *div* adalah faktor pembagi yang sedang diuji.

Berikut adalah penjelasan langkah-langkah yang dilakukan oleh fungsi “*prima*”:

1. Jika *n* kurang dari atau sama dengan 2, maka bilangan tersebut hanya prima jika nilainya adalah 2. Kita mengembalikan hasil perbandingan *n* == 2.
2. Jika *n* dapat dibagi habis oleh *div*, maka *n* bukan bilangan prima. Kita mengembalikan False.
3. Jika *div* yang sedang diuji telah melebihi akar kuadrat dari *n*, maka *n* adalah bilangan prima. Kita mengembalikan True.
4. Jika belum ada kondisi yang terpenuhi, kita melakukan pemanggilan rekursif terhadap fungsi *is_prime* dengan nilai *div* yang ditingkatkan sebesar 1.

Setelah mendefinisikan fungsi “*prima*”, kita dapat menggunakan fungsi tersebut untuk memeriksa suatu bilangan dan memberikan output apakah bilangan tersebut adalah bilangan prima atau bukan.

SOAL 2



```
File Edit Selection View Go Run ... Laprak 12
soal_13_2.py X
D: > Kumpulan tugas kuliah > Alpro > Laprak 13 > soal_13_2.py > ...
1 def is_palindrome(string):
2     string = string.lower().replace(" ", "")
3
4     if len(string) <= 1:
5         return True
6
7     if string[0] == string[-1]:
8         return is_palindrome(string[1:-1])
9     else:
10        return False
11
12 sentence = input("Masukkan sebuah kalimat: ")
13 if is_palindrome(sentence):
14     print("Kalimat tersebut adalah palindrom.")
15 else:
16     print("Kalimat tersebut bukan palindrom.")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "d:/Kumpulan tugas kuliah/Alpro/Laprak 13/soal_13_2.py"
y
Masukkan sebuah kalimat: APA
Kalimat tersebut adalah palindrom.
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "d:/Kumpulan tugas kuliah/Alpro/Laprak 13/soal_13_2.py"
y
Masukkan sebuah kalimat: MANTAP
Kalimat tersebut bukan palindrom.
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12>
```

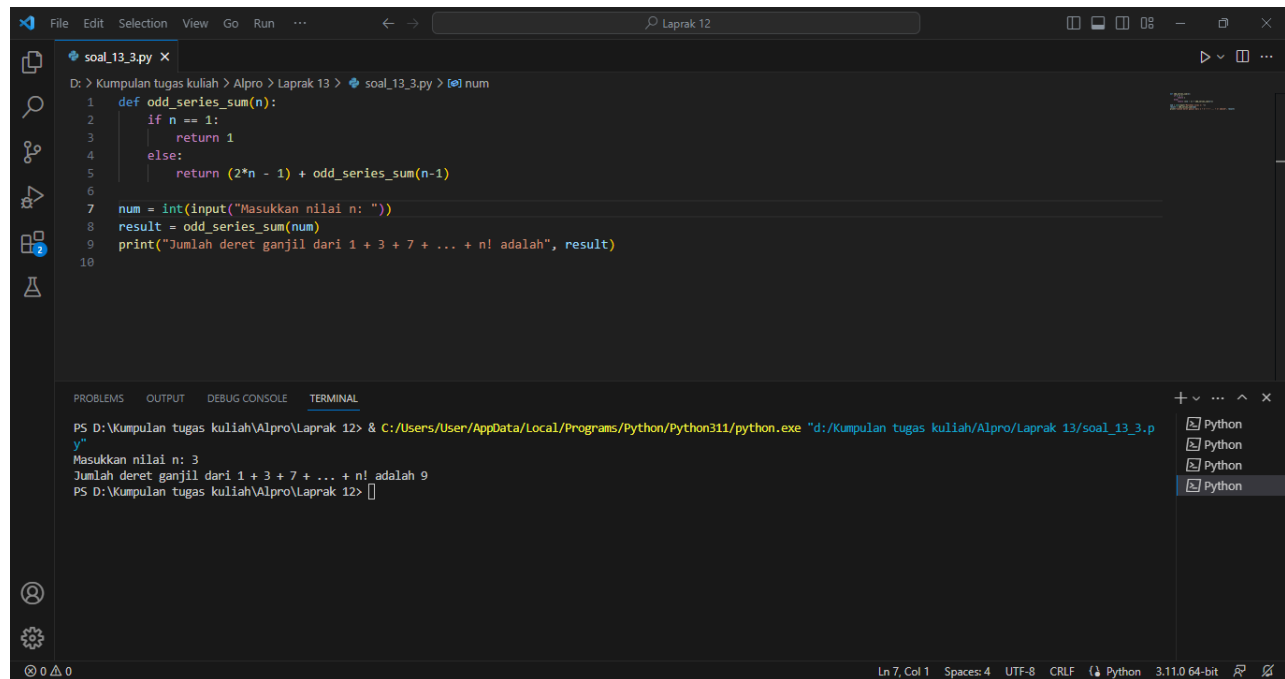
Dalam program di atas, kita menggunakan fungsi rekursif `is_palindrome` untuk mengecek apakah suatu kalimat string adalah palindrom atau bukan.

Berikut adalah penjelasan langkah-langkah yang dilakukan oleh fungsi `is_palindrome`:

1. Mengubah semua huruf dalam kalimat menjadi huruf kecil menggunakan metode `lower()`.
2. Menghapus semua spasi dalam kalimat menggunakan metode `replace(" ", "")`.
3. Jika panjang kalimat kurang dari atau sama dengan 1, maka kalimat tersebut dianggap palindrom. Kita mengembalikan `True`.
4. Jika huruf pertama dan terakhir dalam kalimat sama, maka kita memanggil fungsi `is_palindrome` dengan memotong kedua huruf tersebut dari kalimat menggunakan slicing `string[1:-1]`.
5. Jika huruf pertama dan terakhir dalam kalimat tidak sama, maka kalimat tersebut bukan palindrom. Kita mengembalikan `False`.

Setelah mendefinisikan fungsi `is_palindrome`, kita dapat menggunakan fungsi tersebut untuk memeriksa apakah suatu kalimat adalah palindrom atau bukan.

SOAL 3



```
1 def odd_series_sum(n):
2     if n == 1:
3         return 1
4     else:
5         return (2*n - 1) + odd_series_sum(n-1)
6
7 num = int(input("Masukkan nilai n: "))
8 result = odd_series_sum(num)
9 print("Jumlah deret ganjil dari 1 + 3 + 7 + ... + n! adalah", result)
10
```

PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "d:/Kumpulan tugas kuliah/Alpro/Laprak 13/soal_13_3.py"

Masukkan nilai n: 3

Jumlah deret ganjil dari 1 + 3 + 7 + ... + n! adalah 9

PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12>

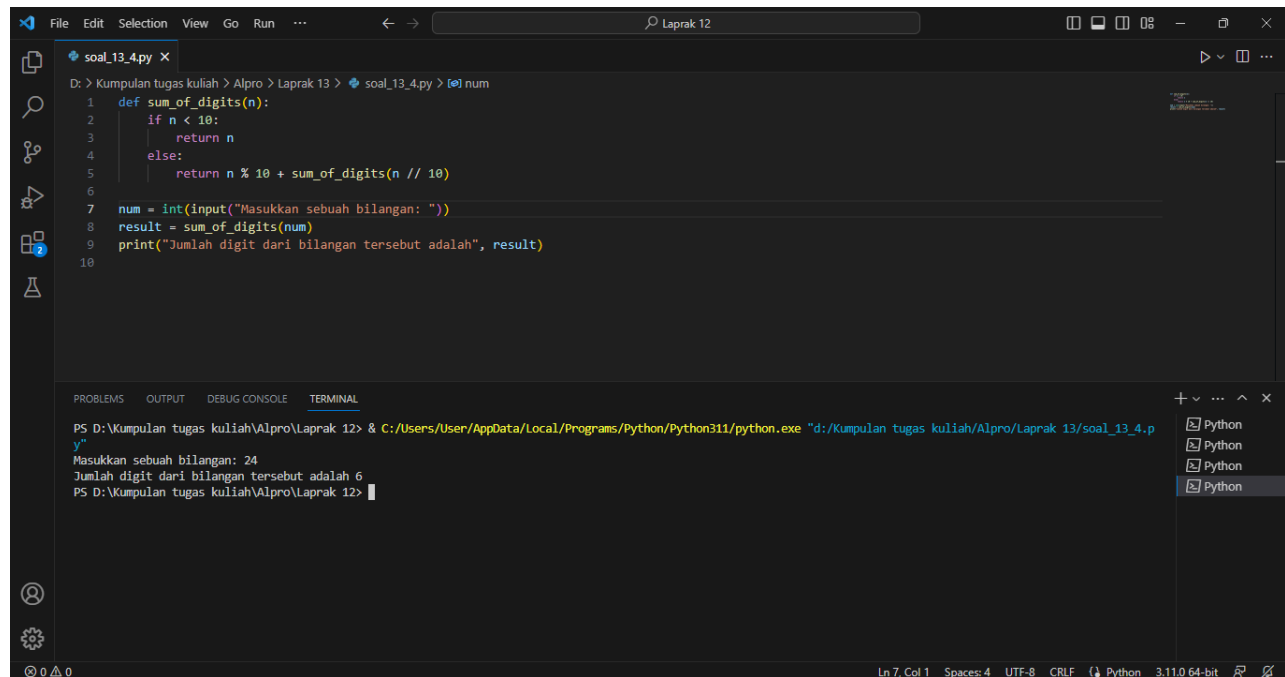
Dalam program di atas, kita menggunakan fungsi rekursif `odd_series_sum` untuk menghitung jumlah deret ganjil dari $1 + 3 + 7 + \dots + n!$.

Berikut adalah penjelasan langkah-langkah yang dilakukan oleh fungsi `odd_series_sum`:

1. Jika n sama dengan 1, maka deret tersebut hanya memiliki satu angka ganjil, yaitu 1. Kita mengembalikan 1.
2. Jika n lebih besar dari 1, maka kita menghitung jumlah deret ganjil dengan rumus $(2n - 1) + \text{odd_series_sum}(n-1)$. Bagian $(2n - 1)$ adalah suku ke- n dalam deret ganjil, sedangkan `odd_series_sum(n-1)` adalah jumlah deret ganjil dari $1 + 3 + 7 + \dots + (n-1)!$ (rekursi).
3. Kita melakukan pemanggilan rekursif terhadap fungsi `odd_series_sum` dengan nilai n yang dikurangi 1.

Setelah mendefinisikan fungsi `odd_series_sum`, kita dapat menggunakan fungsi tersebut untuk menghitung jumlah deret ganjil dari $1 + 3 + 7 + \dots + n!$.

SOAL 4



```
soal_13_4.py X
D: > Kumpulan tugas kuliah > Alpro > Laprak 13 > soal_13_4.py > num
1 def sum_of_digits(n):
2     if n < 10:
3         return n
4     else:
5         return n % 10 + sum_of_digits(n // 10)
6
7 num = int(input("Masukkan sebuah bilangan: "))
8 result = sum_of_digits(num)
9 print("Jumlah digit dari bilangan tersebut adalah", result)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe "d:/Kumpulan tugas kuliah/Alpro/Laprak 13/soal_13_4.py"
Masukkan sebuah bilangan: 24
Jumlah digit dari bilangan tersebut adalah 6
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12>
```

Dalam program di atas, kita menggunakan fungsi rekursif `sum_of_digits` untuk menghitung jumlah digit dari suatu bilangan `n`.

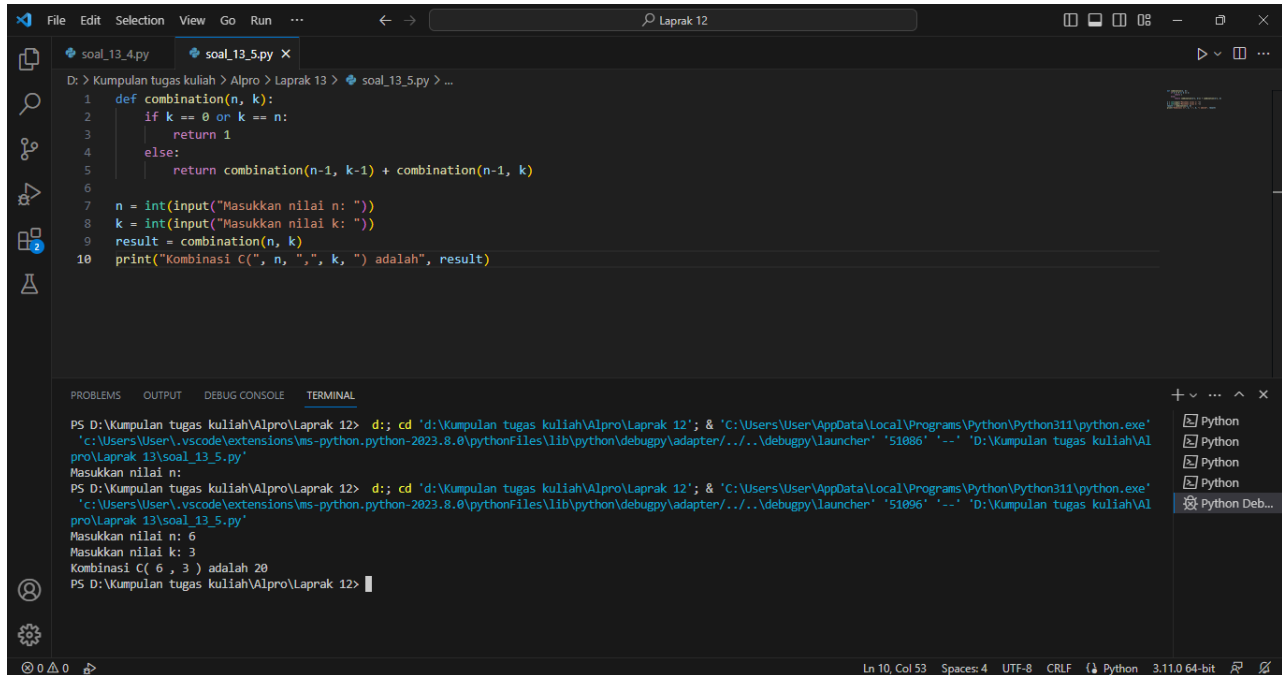
Berikut adalah penjelasan langkah-langkah yang dilakukan oleh fungsi `sum_of_digits`:

1. Jika bilangan `n` kurang dari 10, artinya bilangan tersebut hanya memiliki satu digit. Kita mengembalikan nilai `n`.
2. Jika bilangan `n` lebih dari atau sama dengan 10, maka kita menggunakan operasi modulo (%) untuk mendapatkan digit terakhir bilangan tersebut, yaitu `n % 10`. Kemudian kita memanggil rekursif fungsi `sum_of_digits` dengan bilangan `n` dibagi 10 menggunakan operasi floor division (//) untuk memperoleh bilangan yang lebih kecil tanpa digit terakhir.
3. Kita menjumlahkan digit terakhir dengan hasil rekursif yang didapatkan dari bilangan yang lebih kecil.

Setelah mendefinisikan fungsi `sum_of_digits`, kita dapat menggunakan fungsi tersebut untuk menghitung jumlah digit dari suatu bilangan.

SOAL 5

Untuk menghitung kombinasi menggunakan fungsi rekursif dalam bahasa Python, kita dapat menggunakan rumus matematis kombinasi, yang dikenal sebagai "n choose k" atau " $C(n, k)$ ". Berikut adalah contoh programnya:



```
File Edit Selection View Go Run ... Laprak 12
soal_13_4.py soal_13_5.py X
D:\Kumpulan tugas kuliah\Alpro> Laprak 13 > soal_13_5.py > ...
1 def combination(n, k):
2     if k == 0 or k == n:
3         return 1
4     else:
5         return combination(n-1, k-1) + combination(n-1, k)
6
7 n = int(input("Masukkan nilai n: "))
8 k = int(input("Masukkan nilai k: "))
9 result = combination(n, k)
10 print("Kombinasi C(", n, ", ", k, ") adalah", result)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> d:; cd 'd:\Kumpulan tugas kuliah\Alpro\Laprak 12'; & 'C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe'
'c:\Users\User\.vscode\extensions\ms-python.python-2023.8.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '51886' '--' 'D:\Kumpulan tugas kuliah\Al
pro\Laprak 13\soal_13_5.py'
Masukkan nilai n:
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> d:; cd 'd:\Kumpulan tugas kuliah\Alpro\Laprak 12'; & 'C:\Users\User\AppData\Local\Programs\Python\Python311\python.exe'
'c:\Users\User\.vscode\extensions\ms-python.python-2023.8.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '51896' '--' 'D:\Kumpulan tugas kuliah\Al
pro\Laprak 13\soal_13_5.py'
Masukkan nilai n: 6
Masukkan nilai k: 3
Kombinasi C( 6 , 3 ) adalah 20
PS D:\Kumpulan tugas kuliah\Alpro\Laprak 12> |
```

Dalam program di atas, kita menggunakan fungsi rekursif combination untuk menghitung kombinasi "n choose k" atau " $C(n, k)$ ".

Berikut adalah penjelasan langkah-langkah yang dilakukan oleh fungsi combination:

1. Jika nilai k sama dengan 0 atau k sama dengan n, maka kombinasi tersebut adalah 1. Kita mengembalikan nilai 1.
2. Jika k dan n bukan 0 atau n, maka kita menggunakan rumus kombinasi: $C(n, k) = C(n-1, k-1) + C(n-1, k)$. Pemanggilan rekursif dilakukan untuk dua kasus: pertama dengan n dikurangi 1 dan k dikurangi 1, dan kedua dengan n dikurangi 1 dan k tetap.
3. Kita menjumlahkan hasil rekursif dari dua kasus tersebut.

Setelah mendefinisikan fungsi combination, kita dapat menggunakan fungsi tersebut untuk menghitung kombinasi antara dua nilai n dan k.

